1. Improved DBSCAN

DBSCAN is a classic density-based clustering method that has the advantage of not requiring a preset number of clusters, and it can find any number or shape of clusters in a data set containing noise. The main idea of the algorithm is to examine the regional density of the sample according to the parameters. If the density of any region is higher than the threshold, the region can be regarded as a cluster, and then continue to grow. Assuming the data set is , several key definitions are as follows:

Eps-neighborhood: Given a radius Eps, the Eps-neighborhood of an object p contains every object whose distance from p is not greater than the Eps.

core object: If an object p has at least MinPts objects in its Eps-neighborhood, then p is the core object, and MinPts can be called threshold.

directly density-reachable: For a core object p, it is directly density-reachable to all objects in the Eps-neighborhood.

The classic DBSCAN algorithm searches for clusters by calculating whether the data objects in the neighborhood satisfy a given threshold. When the number of data objects in the neighborhood of p is greater than or equal to MinPts, a new cluster with p as the core object is formed. Subsequently, iteratively checks each object that can be directly density-reachable by the core object to determine whether it satisfies the condition to grow the cluster. When no unvisited object is reachable, the algorithm ends.

However, the algorithm needs to artificial set parameters (Eps and MinPts) without prior knowledge, and the value of the parameters will directly affect the clustering results. The determination of parameters often requires a large number of comparative experiments and expert analysis, which will result in time consumption and waste of resources.

In order to improve the quality of clustering, reduce resource consumption, and overcome the sensitivity of input parameters, we introduce the idea of k-Nearest Neighbor into DBSCAN clustering algorithm, and replace Eps and MinPts by input K. When calculating the distance between two data objects, a weighted distance based on information entropy is used. The following are related definitions:

k-distance: For any positive integer k, the k-distance of an object p, denoted as , is defined as the distance between p and an object :

1. for at least k objects it holds that , and
2. for at most k-1 objects it holds that

k-distance neighborhood: Given the k-distance of an object p, k-distance neighborhood of p contains every object whose distance from p is not greater than the k-distance.

In the improved algorithm, the first thing to do is to traverse the data set and calculate k-distance for each data object. Subsequently, the mean value of the k-distance of all data objects, denoted as , is calculated and the value is set as a new threshold value.

core object (new): For any , if , then p is the core object.

directly density-reachable (new): For a core object p, it is directly density-reachable to all objects in its k-distance neighborhood.

The k-distance of objects in regions with higher density is generally smaller, and the k-distance of objects in regions with lower density is generally larger. In the improved algorithm, is obtained as the threshold value through the overall distribution of the data, only the object with higher density can be set as the core object, which is consistent with the core idea of the original DBSCAN algorithm. However, in the improved algorithm, only the parameter k is required to be input, and the cluster radius is determined by k-distance, which reduces the amount of experiments.

In order to improve the quality of outlier detection, the weighted distance is used to calculate the k-distance of the two objects, and the information entropy is used to determine the weight of the weighted distance. For a d-dimensional data set D，which attribute set is , The value on the attribute of the object in D is written as .

Outlier attribute: For , If the following relationship is satisfied, the attribute is called an outlier attribute.

weighted distance: For , let be the weight of the jth dimension, then the weighted distance between p and q is:

Algorithm description:

Input：, k

1. Traverse the data set , and compute for every object in D.
2. Calculate the , and set it as the threshold value.
3. Traverse the data set again, for , if is satisfied, add to the core object set
4. Randomly select a core object , find all the points that is density-reachable from o, then a cluster can be generated.
5. Similar to the original DBSCAN, the algorithm will stop until all core objects are accessed.

Output: clustering result , set of outliers not divided into any cluster , and number of outliers

1. Improved LOF

Local Outlier Factor (LOF) algorithm give each object a degree of being an outlier, and the degree depends on how isolated the object is with respect to the surrounding neighborhood. Definitions of k-distance and k-distance neighborhood are also used in LOF algorithm. In addition to the above two, the following will introduce the other basic concepts in the LOF algorithm briefly.

Reachability distance: Reachability distance of an object p with respect to object o is defined as:

Local reachability density: The local reachability density of object p is defined as；

Local outlier factor: The Local outlier factor of an object p is defined as:

Since the outliers account for a small proportion in the dataset, directly calculating the outliers of the data objects will increase the amount of calculation. In order to solve this problem, we have improved the original algorithm. Based on k-distance and k-distance neighborhood, this paper proposes a new definition: k-neighborhood density, which is expressed as follows

Based on the output of the improved DBSCAN algorithm, we calculate the k-neighborhood density for each object in the dataset. Then, for each cluster, the k-neighborhood density of the objects in the cluster is arranged in ascending order. This is because the higher the k-neighbor density of an object, the denser the distribution of the region where the object located, the less likely the object is an outlier, so there is no need to calculate the LOF value of the object. Objects with lower k-neighborhood density are more likely to be distributed at the boundary of the cluster. Therefore, the uncertainty of whether these objects are outliers is relatively higher, and further calculation is needed by calculating the LOF value.

At this point, the data objects that need our attention mainly include two parts: outliers obtained by the improved DBSCAN algorithm, and objects that are classified into a cluster but have a small k-neighborhood density. Set a ratio , let . That is, only the first objects with lower density are selected in each cluster. Put selected objects from each cluster and outliers set into a new data set, denoted as D', as the input for the improved LOF algorithm. The above can improve the execution efficiency of the LOF algorithm while ensuring the accuracy of the results, and effectively reduce the amount of calculation. The specific flow of the algorithm is as follows



In the improved LOF algorithm, the obtained LOF values are sorted in descending order, and the top objects (consistent with the number of outliers output by the improved DBSCAN algorithm) are output for subsequent processing, denoted as

1. DBSCAN+LOF



In the DBSCAN+LOF algorithm, we first use the improved DBSCAN to perform a relatively rough analysis of the data set to obtain a clustering result and an outlier set. Subsequently, the improved LOF algorithm is used to screen out objects located at the cluster boundary, or objects that are more likely to be an outlier. The LOF values of those objects are further calculated, and the same number of outliers as the improved DBSCAN obtained are output. Finally, take the union of the two outliers sets obtained in the above steps to obtain the final set of outliers.